

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Methods And Apparatuses For Identifying Remote And
Local Services**

Inventor(s):

David Ethan Zoller

Adam J. Overton

Nadim Abdo

1 **TECHNICAL FIELD**

2 This invention relates to computers and more particularly to methods and
3 apparatuses for identifying to a user of a local computing device that certain local
4 computing device based services and/or remote computing device based services
5 are providing the graphical user interface (GUI) shown on a local display
6 screen(s).

7
8 **BACKGROUND OF THE INVENTION**

9 Most of the personal computers (PCs) and other like computing devices in
10 service today present the user with an interactive graphical user interface (GUI) on
11 a display screen. One type of GUI, which is associated with operating systems, is
12 a "desktop" GUI. Through a desktop GUI a user can interactively select graphical
13 icons, etc., as needed to use the computer's resources. For example, the user
14 might selectively activate a program application, peruse the contents of a data
15 storage device, or perhaps initiate communication with a remotely located
16 computing device. Many of the selections may themselves cause other GUIs to be
17 opened on the desktop and displayed.

18 There is a movement underway to provide locally located users with the
19 ability to be presented with (and interact with) a desktop or other GUI that is
20 actually associated with another computing device that is remotely located, but
21 operatively connected to a local computing device.

22 Thus, for example, a user that is located away from their business' PC may
23 nevertheless access the business PC desktop through another PC, such as a home
24 PC that is configured to communicate with the business' PC. Here, the user
25 initiates a remote desktop connection through some program. By way of example,

1 in certain exemplary operating systems a Terminal Services Client program may
2 be used. The remote desktop connection program essentially opens up the
3 business PC's desktop GUI on the home PC's display screen.

4 In this situation and others like it, many users prefer to have the remote
5 desktop GUI opened up fully on the display screen, such that it more closely
6 resembles the desktop GUI that they normally work with. Thus, in the previous
7 example, the user at home would see his/her business PC's desktop as if he/she
8 were actually at work rather than at home. Providing a full remote desktop display
9 not only provides familiarity to the user, but also prevents users from being
10 confused by the potential of seeing two objects/icons that appear to be the same
11 but are actually associated with different desktop GUIs. Moreover, a substantially
12 full screen remote desktop GUI takes advantage of often precious display screen
13 real estate.

14 One of the drawbacks to having a full screen remote desktop GUI (or other
15 type of GUI) is that some users may forget which computing device, i.e., local or
16 remote, they are viewing and interacting with. Consequently, inexperienced users
17 may, for example, misplace files and/or quickly become frustrated when they
18 cannot locate certain files/programs.

19 For these reasons and others, there is a need to provide users with a
20 mechanism to identify which desktop or other like substantially full screen GUI
21 they are viewing and/or interacting with.

SUMMARY OF THE INVENTION

Methods and apparatuses are provided which visually identify a desktop or other graphical user interface (GUI) as being associated with a session or other like process operating on a particular computing device.

For example, in accordance with certain aspects of the present invention, a method is provided, which includes displaying a first GUI on a display screen and a second GUI over the first GUI. Here, the first GUI is associated with a program running on a first computing device and the second GUI is associated with a program running a second (e.g., remote) computing device. In accord with the method, the second GUI is displayed on substantially the full screen of the display screen and includes at least one identifier that identifies that the second GUI is not associated with the first computing device. In certain implementations, for example, the first and/or second GUI may include a desktop GUI associated with an operating system, or perhaps an application GUI associated with an application program. In certain other implementations the identifier is selectively displayed for a defined period of time and then no longer displayed until and unless reactivated/re-displayed in some manner either based on some logic (e.g., a timer) and/or perhaps the user's activity within the displayed GUI. In certain further implementations, for example, the identifier may be reactivated/re-displayed based on user keyboard/keypad input(s), special hardware button input(s), and the like. In still other implementations, the identifier may be selectively reactivated/re-displayed by logic operating on the remote computing device.

In accordance with certain further aspects of the present invention, a system is provided which includes a first computing device that is coupled to a display screen and also to a second computing device, e.g., through at least one

interconnecting communication link. The first computing device is configured to display a first GUI on the display screen. Here, the first GUI is associated with a program running on the first computing device. The second computing device is configured to display a second GUI on the display screen over the first GUI. The second GUI is associated with a program operatively configured on the second computing device. To alert the user to that fact, the second GUI includes at least one identifier that identifies that the second GUI is not associated with the first computing device.

The above stated needs and others are also met by an apparatus that includes a computing device. Here, the computing device is capable of being operatively connected to at least one other computing device through an interconnecting communication channel. The computing device is configured with logic that generates GUI data suitable for display on a display screen coupled to the other computing device. If displayed on the display screen, the GUI data is configured to use substantially the full area of the display screen. As such, the GUI data advantageously also includes data for displaying at least one identifier that identifies that the GUI data is associated with the logic of the computing device as opposed to the other computing device.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an exemplary computing device and computing environment capable of presenting a user with a plurality of graphical user interfaces (GUIs) associated with different computing devices and/or sessions, in accordance with certain implementations of the present invention.

1 Fig. 2 is an illustrative diagram depicting the presentation of a second
2 graphical user interface (GUI) over an initial GUI, wherein the second GUI
3 includes at least one identifier that selectively alerts the user that the GUI currently
4 being displayed is the second GUI rather than the initial GUI, in accordance with
5 certain exemplary implementations of the present invention.

6 Fig. 3A is an illustrative screen shot showing a desktop associated with a
7 second graphical user interface (GUI) and an identifier that selectively alerts the
8 user that the GUI currently being displayed is associated with a particular
9 computing device and/or session, in accordance with certain exemplary
10 implementations of the present invention.

11 Fig. 3B illustrates certain features of the exemplary identifier of Fig. 3A, in
12 accordance with certain implementations of the present invention.

13 Fig. 3C is an illustrative screen shot showing an opened substantially full-
14 screen graphical user interface (GUI) of a word processor application program that
15 has further been highlighted to depict one exemplary location within the GUI that
16 an identifier, for example, as in Figs. 2, 3A, and/or 3B, may be located and/or
17 selectively displayed to alert the user that the GUI currently being displayed is
18 associated with a particular computing device and/or session, in accordance with
19 certain exemplary implementations of the present invention.

20 Figs 4A-E are block diagrams illustrating that one or more identifiers may
21 be located at various positions within a graphical user interface (GUI) and
22 configured to selectively alert the user that the GUI currently being displayed is
23 associated with a particular computing device and/or session, in accordance with
24 certain exemplary implementations of the present invention.
25

Fig. 5 is a flow diagram depicting a method for selectively alerting a user that a GUI currently being displayed is associated with a particular computing device and/or session, in accordance with certain exemplary implementations of the present invention.

DETAILED DESCRIPTION

Exemplary Computing Environment:

As shown in Fig. 1, computer 20 includes one or more processors or processing units 21, a system memory 22, and a bus 23 that couples various system components including the system memory 22 to processors 21. Bus 23 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures.

The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within computer 20, such as during start-up, is stored in ROM 24.

Computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from and writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM, DVD ROM or other optical media. The hard disk drive 27, magnetic disk drive 28 and optical disk drive 30 are each connected to bus 23 by applicable interfaces 32, 33 and 34, respectively.

1 The drives and their associated computer-readable media provide
2 nonvolatile storage of computer readable instructions, data structures, program
3 modules and other data for computer 20. Although the exemplary environment
4 described herein employs a hard disk, a removable magnetic disk 29 and a
5 removable optical disk 31, it should be appreciated by those skilled in the art that
6 other types of computer readable media which can store data that is accessible by a
7 computer, such as magnetic cassettes, flash memory cards, digital video disks,
8 random access memories (RAMs) read only memories (ROM), and the like, may
9 also be used in the exemplary operating environment.

10 A number of program modules may be stored on the hard disk, magnetic
11 disk 29, optical disk 31, ROM 24, or RAM 25, including an operating system 35,
12 one or more application programs 36, other program modules 37, and program
13 data 38. A user may enter commands and information into computer 20 through
14 input devices such as keyboard 40 and pointing device 42. Other input devices
15 (not shown) may include a microphone, joystick, game pad, satellite dish, scanner,
16 or the like. These and other input devices are connected to the processing unit 21
17 through an interface 46 that is coupled to bus 23.

18 A monitor 47 or other type of display device is also connected to bus 23 via
19 an interface, such as a video adapter 48. In addition to the monitor, personal
20 computers typically include other peripheral output devices (not shown) such as
21 speakers and printers.

22 Computer 20 can operate in a networked environment using logical
23 connections to one or more remote computers, such as a remote computer 50.
24 Remote computer 50 may be another personal computer, a server, a router, a
25 network PC, a peer device or other common network node, and typically includes

1 many or all of the elements described above relative to computer 20. The logical
2 connections depicted in Fig. 1 include a local area network (LAN) 51 and a wide
3 area network (WAN) 52. Such networking environments are commonplace in
4 offices, enterprise-wide computer networks, intranets, and the Internet.

5 When used in a LAN networking environment, computer 20 is connected to
6 the local network 51 through a network interface or adapter 53. When used in a
7 WAN networking environment, computer 20 typically includes a modem 54 or
8 other means for establishing communications over the wide area network 52, such
9 as the Internet. Modem 54, which may be internal or external, is connected to bus
10 23 via interface 46. In a networked environment, program modules depicted
11 relative to the personal computer 20, or portions thereof, may be stored in the
12 remote memory storage device. It will be appreciated that the network
13 connections shown are exemplary and other means of establishing a
14 communications link between the computers may be used.

16 Exemplary Remote Desktop GUI Having An Identifier Feature:

17 Reference is now made to Fig. 2, which is an illustrative diagram depicting
18 the display of two different graphical user interfaces (GUIs) associated with two
19 different computing devices and/or sessions. This example uses desktop GUIs
20 associated with operating systems. Those skilled in the art will recognize that the
21 methods and apparatuses described herein are also fully applicable to other types
22 of GUIs and in particular GUIs that make use of substantially all of the available
23 display screen area.

24 As shown, a first desktop GUI 202 is the first GUI to be displayed to the
25 user of computer 20 on display 47 (see Fig. 1). Here, first desktop GUI 202 is

1 associated with operating system 35 of PC 20 and an open session running
2 thereon. In this exemplary arrangement, first desktop GUI 202 includes at least
3 one desktop object 204a, a start button 208a and a task bar 210a.

4 In Fig. 2, the user selectively begins a session with remote computer 50,
5 which, as demonstrated in Fig. 1, can be operatively connected to computer 20.
6 This remote session can be accomplished, for example, through various types of
7 terminal server or other like software applications/modules operating on
8 computers 20 and 50; such capabilities are well-known. As a result of this user
9 initiated remote session, a second desktop GUI 210 associated with remote
10 computer 50 is displayed to user on display 47 instead of (e.g., over) first desktop
11 GUI 202. The illustrative arrows depict the switching of desktop GUIs as a result
12 of the remote session being started. Note that the exemplary desktop GUIs are
13 opened in a full screen mode or at least a substantially full screen mode.

14 As illustrated at the bottom of Fig. 2 the resulting second desktop GUI 210,
15 which is associated with operating system or other like software running on
16 remote computer 50, has many similar features to first desktop GUI 202. For
17 example, second desktop GUI 210 may include: one or more desktop objects
18 204b, such as, e.g., a file or folder icon, etc.; a start button 208b to allow the user
19 to initiate some action, such as, e.g., launch a program application, etc.; and, a task
20 bar 210b that depicts certain files, actions, programs, etc., associated with the
21 user's session.

22 If first desktop GUI 202 and second desktop GUI 210 appear too much
23 alike, then the user may be confused about which GUI they are viewing and
24 interacting with and therefore which computer/session they are actually using. To
25 prevent this and other potential problems from occurring, second desktop GUI 210

1 further includes an identifier 212 that alerts the user that second desktop GUI 210
2 is associated with remote computer 50 and/or not computer 20. In this exemplary
3 implementation, identifier 212 is located near the top center of second desktop
4 GUI 210. This is one potential position for identifier 212. In other
5 implementations, identifier 212 may be in other locations and may occupy more
6 than one area within second desktop GUI 210. Some examples are shown in Figs
7 4A-E, wherein a generic GUI 200 includes one or more identifiers 212 arranged at
8 various locations.

9 The location, shape, color, information, and/or any other visual
10 characteristics associated with identifier 212 may depend on the type of GUI in
11 which it is being displayed.

12 Reference is now made to Fig. 3A. In this exemplary screen shot a second
13 desktop GUI 210' is depicted as having an identifier 212 in the form of a
14 "connection bar" which is displayed near the center at the top portion of the
15 desktop space. The connection bar identifier 212 in this example is preferably
16 visibly different than the surrounding GUI displayed area, so the user will
17 hopefully be alerted/aware of the fact that the displayed GUI 210' is associated
18 with a connected session/device and/or not local computing device 20.

19 In accordance with certain further aspects of the present invention, the
20 displaying of identifier 212 may be selectively controlled. For example, in certain
21 implementations the connection bar identifier 212 in Fig. 3A may be displayed for
22 a certain amount of time to begin with and then no longer displayed unless
23 requested by the user and/or otherwise reactivated by the underlying logic. For
24 example, the display of identifier 212 can be timed in some manner to initially be
25 displayed for a defined period of time following the connection and then at least

1 partially if not fully removed from the GUI display. Logic such as this can cause
2 the "hidden" identifier 212 to reactivate/reappear after a specific timed hiatus, for
3 example.

4 In certain implementations, if the user causes a pointing device (e.g., a
5 mouse), to have its cursor pointed in the area or other nearby regions wherein
6 identifier 212 was previously displayed, then logic can cause identifier 212 to be
7 reactivated/redisplayed within GUI 210'. Certain implementations require that the
8 pointer be within the region for a specific amount of time before the hidden
9 identifier reappears. Thus, the identifier can be operatively configured to have an
10 "auto-hide" capability. In other implementations, a more obvious button may be
11 provided within GUI 210' (or perhaps a keystroke defined) to allow the identifier
12 212 to be toggled on/off by the user.

13 These and other like implementations allow the user to initially notice that
14 the GUI is associated with a remote session/device and then have the identifier
15 212 selectively disappear/reappear from the display. In certain implementations
16 however, the ability to hide identifier 212 may itself be controlled. For example,
17 for security or other reasons a system administrator or like person may establish
18 and implement a policy by which a computer or network of computers must
19 continuously or at least periodically display an identifier 212 such that the user(s)
20 are kept aware that they are using a remote computer or like device.

21 In still other exemplary implementations, identifier 212 may include one or
22 more selectable GUI objects and the like. For one example, see Fig. 3B, which
23 depicts the connection bar identifier 212 of Fig. 3A in more detail. Here, within
24 identifier 212 the remote connection is described or otherwise identified by name
25 304 (e.g., device and/or session name, etc.). Also provided is a selectable button

1 302 that is associated with a system menu that the user can view/interface with, a
2 user selectable minimize toggle button 306, a user selectable full screen/tiled
3 toggle button 308, and a user selectable close button 310. These and other like
4 user selectable features can be provided to allow the user more access to the
5 system menus, title bars, and min/max/close controls of applications and the like
6 operating on the remote computer.

7 These are just a few examples of the information/functionality that can be
8 provided with identifier 212. Indeed, in certain implementations, remote computer
9 50 can dynamically change the information within identifier 212 during the
10 session. For example, if remote computer 50 is running on battery power, then
11 information about the battery power available may be displayed. Thus, as the
12 batteries are running high/low on charge, an applicable message may be displayed
13 to the user through identifier 212. Practically any other useful information may be
14 presented through identifier 212.

15 Reference is now made to Fig. 3C, which is an illustrative screen shot of a
16 word processor GUI 320 that may be operatively opened within second desktop
17 GUI 210', for example. Although not shown, the connection bar identifier 212
18 would normally occupy the area in and around dashed-line box 322. The reason
19 connection bar identifier 212 is not shown here is to illustrate that, when
20 displayed, identifier 212 is strategically positioned within a normally vacant
21 region of the word processor program application's title bar 324. Thus, identifier
22 212 is preferably strategically positioned so as to not interrupt the user of the
23 remote computing session.

24 With the various above-described exemplary implementations and others
25 like them in mind, Fig. 5 illustrates an exemplary method 500 for displaying an

1 identifier 212 in a generic remote session GUI 200, e.g., a desktop, application,
2 and/or other like program module generated GUI.

3 In step 502, a first GUI is established and displayed by a local
4 computer/session. In step 504, the user is allowed to initiate a session/connection
5 to a remote computer, for example, using the first GUI. Next, in step 506, as a
6 result of step 504 a second GUI is established by a remote computer/session and
7 displayed by the local computer on its display screen. In step 508, selectively
8 displayed within the second GUI is at least one identifier or like object that alerts
9 the user that the second GUI is not associated with the local computer/session but
10 rather it is associated with the remote computer/session.

11 In accordance with certain other aspects of the present invention, an
12 identifier as described herein may also be employed to serve the user of a
13 computer that is capable of having a local application or other like process GUI
14 operate in a full screen or substantially full screen mode, such that another
15 underlying local application or process GUI is essentially hidden from the user.
16 Thus, for example, a local Internet browser application or word processing
17 application may be displayed in full screen or substantially full screen mode over
18 the local computer's operating system desktop. Here, the identifier included in the
19 overlying GUI can be configured, for example, to alert the user that one or more
20 underlying GUIs are also available. To assist the user in switching between
21 various GUIs (local and/or remote operating), the identifier may include a user
22 selectable inputs to open/close/min./max. or perhaps switch the GUIs at will. The
23 identifier, which may have auto-hiding capabilities, may also be useful in
24 identifying the program, file, and/or other information associated with the GUI.
25 Thus, in certain implementations, the identifier may be selectively

1 located/displayed in a manner that advantageously reduces interference with the
2 GUI operation. This could be an improvement to the more traditional title bars,
3 etc., that may reduce the effectiveness of the GUI operation.

4 Although some preferred implementations of the various methods and
5 arrangements of the present invention have been illustrated in the accompanying
6 Drawings and described in the foregoing Detailed Description, it will be
7 understood that the invention is not limited to the exemplary implementations
8 disclosed, but is capable of numerous rearrangements, modifications and
9 substitutions without departing from the spirit of the invention as set forth and
10 defined by the following claims.